REMARKS

Claims 1-6, 8-11, 13-15, and 17-33 are currently pending in the application. Of these claims, claims 1, 10, 20, and 28 are independent.

Rejections of Claims 1-6 and 20-30 Involving Batten

Claims 1-6 and 20-21 are rejected under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent No. 6,260,189 B1 to Batten et al. ("Batten").

Claims 22 and 28-30 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Batten.

Claims 23-24 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Batten in view of U.S. Patent No. 6,427,204 B1 to Arimilli et al. ("Arimilli").

Claims 25-26 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Batten in view of U.S. Patent No. 6,018,798 to Witt et al. ("Witt").

Claim 27 is rejected under 35 U.S.C. § 103(a) as being unpatentable over Batten in view of Rotenberg et al., "A Trace Cache Microarchitecture and Evaluation," 1998 ("Rotenberg").

Applicant respectfully traverses these rejections as follows.

Independent claim 1 recites a dependency descriptor including an address of an instruction sequence.

Independent claims 20 and 28 recite a dependency descriptor including an address of a set of instructions.

Applicant respectfully submits none of the references cited in these rejections, whether alone or in any combination with one another, taught or suggested such features as claimed. More particularly, Applicant respectfully submits the numInstr value in Figure 7 of Batten is not an address.

Noting claims 2-6, 21-27, and 29-30 depend from independent claim 1, 20, or 28, Applicant therefore respectfully submits these rejections have been overcome and should accordingly be withdrawn.

Note that there may be additional reasons for the patentability of claims. For example, there may be additional reasons why the dependent claims are patentable.

Rejections of Claims 10-11, 13-15, and 17-19

Claims 10-11, 13-15, and 17-19 are rejected under 35 U.S.C. § 102(b) as being anticipated by Ranganathan et al., "The PEWs Microarchitecture: Reducing Complexity Through Data-Dependence Based Decentralization," 1998 ("Ranganathan").

Applicant respectfully traverses these rejections as follows.

Applicant respectfully submits Ranganathan did not teach or suggest a trace descriptor including a plurality of dependency descriptors having locations of corresponding instruction sequences as claimed in independent claim 10.

Noting claims 11, 13-15, and 17-19 depend from independent claim 10, Applicant therefore respectfully submits these rejections have been overcome and should accordingly be withdrawn.

Note that there may be additional reasons for the patentability of claims. For example, there may be additional reasons why the dependent claims are patentable.

Rejections of Claims 1-6 and 8-9 Involving Ranganathan

Claims 1-6 and 8-9 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Ranganathan in view of U.S. Patent No. 6,304,962 B1 to Nair ("Nair"). Also, U.S. Patent No. 5,732,255 to Verbauwhede ("Verbauwhede") is cited as extrinsic evidence.

Applicant respectfully traverses these rejections as follows.

Independent claim 1 recites a dependency descriptor including an address of an instruction sequence.

The Office Action on page 10 cites Nair that taught in column 10 at lines 8-17:

> FIG. 6 is a flow chart of a method for prefetching superblocks in a computer processing system according to another embodiment of the present invention. In particular, the method of FIG. 6 corresponds to the STB of FIG. 5. It is to be appreciated that the method of FIG. 6 attempts to predict longer paths of execution than the method of FIG. 3. Unlike a trace-cache scheme or the DIF approach, this scheme does not store the

actual instructions along the path, only the addresses from where the instructions should be fetched.

The Office Action on page 10 also cites Verbauwhede that taught in column 2 at lines 43-49:

> Program memory 4 is typically a single port ROM comprising an array of rows and columns of memory cells. Each row of cells (storage locations) is indexed by a row address. Each row address (sometimes denoted as an "instruction address") is typically a 16-bit word, and each row typically consists of 32 cells (so that each row stores a 32-bit instruction or other 32-bit word).

The Office Action on page 10 concludes:

> * * * A person of ordinary skill in the art would have recognized that by storing the location of the instruction sequence as opposed to the instruction sequence itself, less space would be consumed in the trace cache. In addition, it should be realized that it is not necessarily important whether instructions or instruction addresses (locations) are stored in a trace cache. Instead, all that is required is that instructions may be identified by the trace cache and this may be done by providing the instruction or its address. As a result, it would have been obvious to one or ordinary skill in the art at the time of the invention to modify Ranganathan such that the trace cache (more specifically, the dependency descriptor) stores an instruction location as opposed to the instruction sequence itself.

Applicant initially notes that Nair explicitly taught, as emphasized above, that the superblock prefetching method of Figure 6 is "[u]nlike a trace-cache scheme". Applicant therefore respectfully submits that Nair plainly taught away from combining the superblock prefetching method of Figure 6 in any manner with a trace cache scheme. Because Ranganathan taught a trace cache scheme, Applicant respectfully submits one of ordinary skill in the art would not have been motivated to modify Ranganathan using Nair as otherwise suggested in the Office Action.

Applicant also respectfully submits that, in teaching that the superblock prefetching method of Figure 6 is "[u]nlike a trace-cache scheme", Nair taught that the superblock prefetching method of Figure 6 is a substitute for a trace cache scheme. Applicant respectfully submits, however, that neither Nair nor Ranganathan taught a dependency descriptor including an address of an instruction sequence.

Applicant further respectfully submits that one of ordinary skill in the art would not have been motivated to store instruction addresses instead of instructions in the Trace Cache of Ranganathan because doing so would have been incompatible with the trace cache scheme of Ranganathan. Ranganathan taught, with reference to Figure 3, that instructions fetched from an Icache are decoded by a Decoder and analyzed by a Register Data Flow Graph (RDFG) Analyzer to generate data dependence information. The decoded instructions and data dependence information are stored in the Trace Cache for future accesses. See Ranganathan in sections 3.1 to 3.3. If instruction addresses instead of decoded instructions were stored in the Trace Cache of Ranganathan, then even recently used instructions would have to be fetched again from the Icache, decoded again by the Decoder, and analyzed again by the RDFG Analyzer. This would appear to obviate use of the Trace Cache in Ranganathan.

Noting claims 2-6 and 8-9 depend from independent claim 1, Applicant therefore respectfully submits these rejections have been overcome and should accordingly be withdrawn.

Note that there may be additional reasons for the patentability of claims. For example, there may be additional reasons why the dependent claims are patentable.

New Claims 31-33

New claims 31-33 depend from independent claim 1. Applicant therefore respectfully submits new claims 31-33 are patentable.

It is respectfully submitted this patent application is in condition for allowance, for which early action is earnestly solicited.

The Examiner is invited to telephone the undersigned to help expedite the prosecution of this patent application.

Respectfully submitted,

Date: October 1, 2005

/Matthew C. Fagan, Reg. No. 37,542/

Matthew C. Fagan
Registration No. 37,542
Telephone: (512) 732-3936
Facsimile: (512) 732-3912 (please call first)

c/o Blakely, Sokoloff, Taylor & Zafman, L.L.P.
12400 Wilshire Boulevard, Seventh Floor
Los Angeles, California 90025
Telephone: (503) 439-8778
Telephone: (310) 207-3800